

JsonP

一、 JsonP 简介

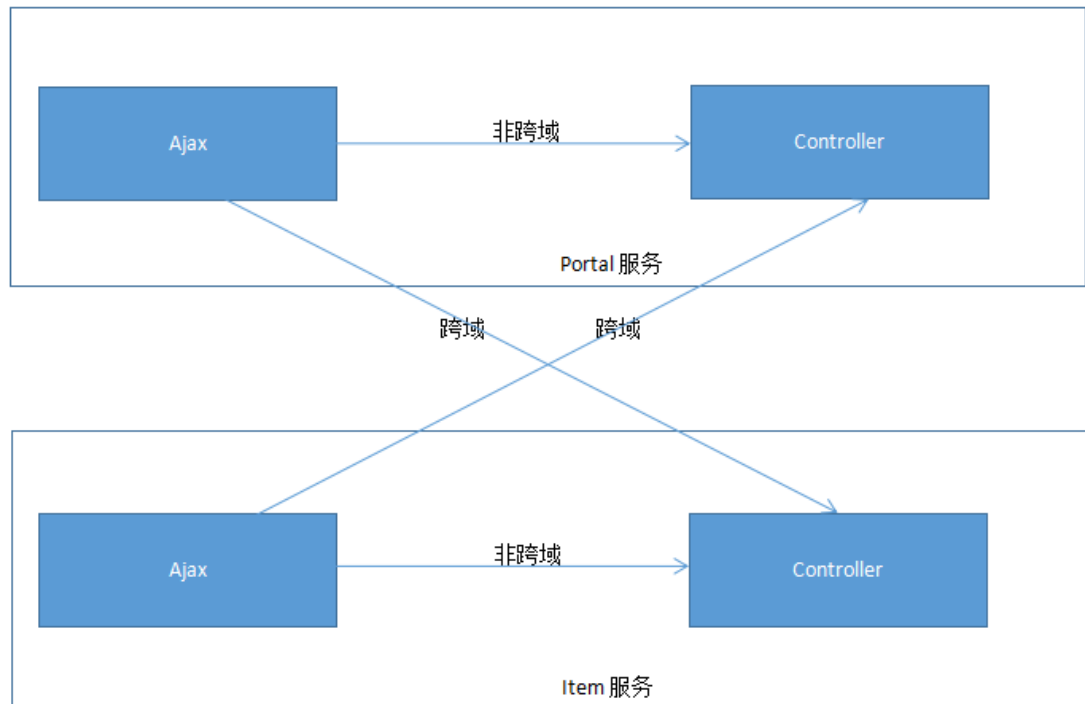
1 什么是 JsonP

Jsonp(JSON with Padding) 是 json 的一种"使用模式", 可以让网页从别的域名(网站)那获取资料, 即跨域读取数据。

为什么我们从不同的域(网站)访问数据需要一个特殊的技术(JSONP)呢? 这是因为同源策略。

2 什么是跨域?

跨域是指一个域(网站)下的文档或脚本试图去请求另一个域(网站)下的资源。



3 什么是同源策略?

同源策略/SOP (Same origin policy) 是一种约定, 由 Netscape 公司 1995 年引入浏览器, 它是浏览器最核心也最基本的安全功能, 现在所有支持 JavaScript 的浏览器都会使用这个策略。如果缺少了同源策略, 浏览器很容易受到 XSS、CSFR 等攻击。所谓同源是指"协议+域名+端口"三者相同, 即便两个不同的域名指向同一个 ip 地址, 也非同源。

3.1 非同源策略限制以下几种行为:

- 1.) Cookie、LocalStorage 和 IndexDB 无法读取

-
- 2.) DOM 和 Js 对象无法获得
 - 3.) AJAX 请求不能发送

4 常见跨域场景

URL	说明	是否允许通信
http://www.domain.com/a.js	同一域名，不同文件或路径	允许
http://www.domain.com/b.js		
http://www.domain.com/lab/c.js		
http://www.domain.com:8000/a.js	同一域名，不同端口	不允许
http://www.domain.com/b.js		
http://www.domain.com/a.js	同一域名，不同协议	不允许
https://www.domain.com/b.js		
http://www.domain.com/a.js	域名和域名对应相同 ip	不允许
http://192.168.4.12/b.js		
http://www.domain.com/a.js	主域相同，子域不同	不允许
http://x.domain.com/b.js		
http://domain.com/c.js		
http://www.domain1.com/a.js	不同域名	不允许
http://www.domain2.com/b.js		

5 跨域解决方案

- 1) 通过 jsonp 跨域
- 2) document.domain + iframe 跨域
- 3) location.hash + iframe
- 4) window.name + iframe 跨域
- 5) postMessage 跨域
- 6) 跨域资源共享 (CORS)
- 7) nginx 代理跨域
- 8) nodejs 中间件代理跨域
- 9) WebSocket 协议跨域

6 JsonP 优缺点

JSONP 的优点是：它不像 XMLHttpRequest 对象实现的 Ajax 请求那样受到同源策略的限制；它的兼容性更好，在更加古老的浏览器中都可以运行，不需要 XMLHttpRequest 或 ActiveX 的支持；并且在请求完毕后可以通过调用 callback 的方式回传结果。

JSONP 的缺点则是：它只支持 GET 请求而不支持 POST 等其它类型的 HTTP 请求；它

只支持跨域 HTTP 请求这种情况，不能解决不同域的两个页面之间如何进行 JavaScript 调用的问题。

二、 JsonP 的使用

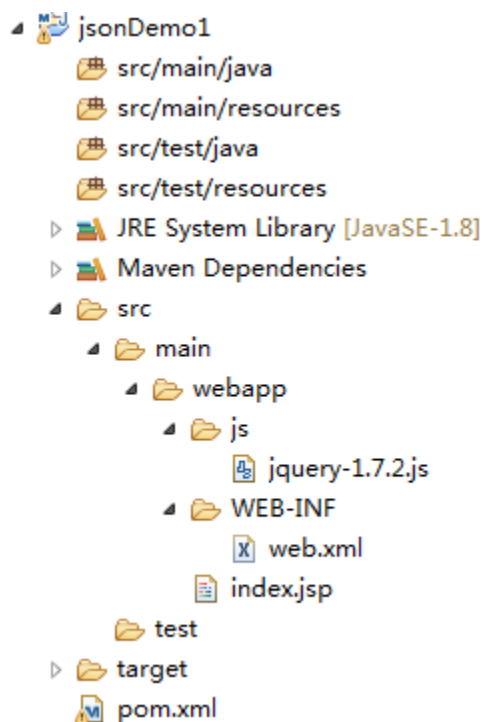
1 搭建跨域场景

1.1 需求：

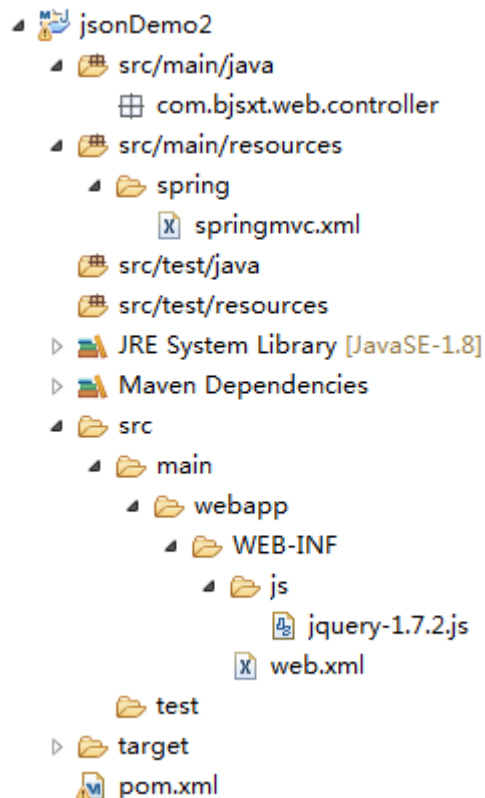
- 1) 创建两个 web 工程，名称为 jsonDemo1(8080)、jsonDemo2(9090)
- 2) jsonDemo1 中提供一个 index.jsp。
- 3) 在 jsonDemo1 的 index.jsp 中通过 Jquery 的 Ajax 跨域请求 jsonDemo2
- 4) jsonDemo2 中使用 springMVC 处理请求，返回一个 json 对象
- 5) 在 jsonDemo1 中将返回的结果插入到 index.jsp 中

1.2 创建项目

1.2.1 jsonDemo1



1.2.2jsonDemo2



2 使用 JsonP 解决跨域

2.1 Ajax 跨域请求时会出现异常

Failed to load <http://localhost:8080/user/findUser>: No 'Access-Control-Allow-Origin' header is present on the requested resource. Origin 'http://localhost:8080' is therefore not allowed access. [index.js:1](#)

2.2 在 ajax 中请求方式有所改变

```
<%@ page language="java" contentType="text/html;
charset=UTF-8"

    pageEncoding="UTF-8"%>

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01
Transitional//EN"
```

```
"http://www.w3.org/TR/html4/loose.dtd">

<html>

<head>

<meta http-equiv="Content-Type"
content="text/html; charset=UTF-8">

<title>Insert title here</title>

<script type="text/javascript"
src="/js/jquery-1.7.2.js"></script>

<script type="text/javascript">

$(function(){

    $("#but").click(function(){

        $.ajax({

            type:"get",

            url:"http://localhost:9090/user/findUser",

            dataType:"jsonp",

            jsonp:"callback",

            success:function(data){

                alert(data);

                var str = "";

                for(i=0;i<data.length;i++){

                    str+= data[i].userid+"

```

```
" + data[i].username + " " + data[i].userage + " ";

        }

        $("#show").html(str);

    }

    });

});

</script>

</head>

<body>

    <span id="show"></span>

    <input type="button" value="OK" id="but"/>

</body>

</html>
```

2.3 请求的 Controller 需要改变

```
@Controller

@RequestMapping("/user")

public class UserController {

    @RequestMapping("/findUser")

    @ResponseBody
```

```
public String findUser(String callback){
    Users user = new Users(1, "admin", 20);
    Users user1 = new Users(2, "zhangsan", 22);
    Users user2 = new Users(3, "lisi", 24);
    List<Users> list = new ArrayList<>();
    list.add(user);
    list.add(user1);
    list.add(user2);
    String json = JsonUtils.objectToJson(list);
    //callback({id:1,name:zhangsan...})
    return callback+"("+json+")";
}
}
```

3 SpringMVC 对 JsonP 的支持

```
@Controller
@RequestMapping("/user")
public class UserController {

    @RequestMapping("/findUser")
    @ResponseBody
```

```
public Object findUser(String callback){
    Users user = new Users(1, "admin", 20);
    Users user1 = new Users(2, "zhangsan", 22);
    Users user2 = new Users(3, "lisi", 24);
    List<Users> list = new ArrayList<>();
    list.add(user);
    list.add(user1);
    list.add(user2);
    //json 转换
    MappingJacksonValue mv = new
MappingJacksonValue(list);
    mv.setJsonpFunction(callback);
    return mv;
}
}
```

注意：需要依赖 Jackson 的 jar 包